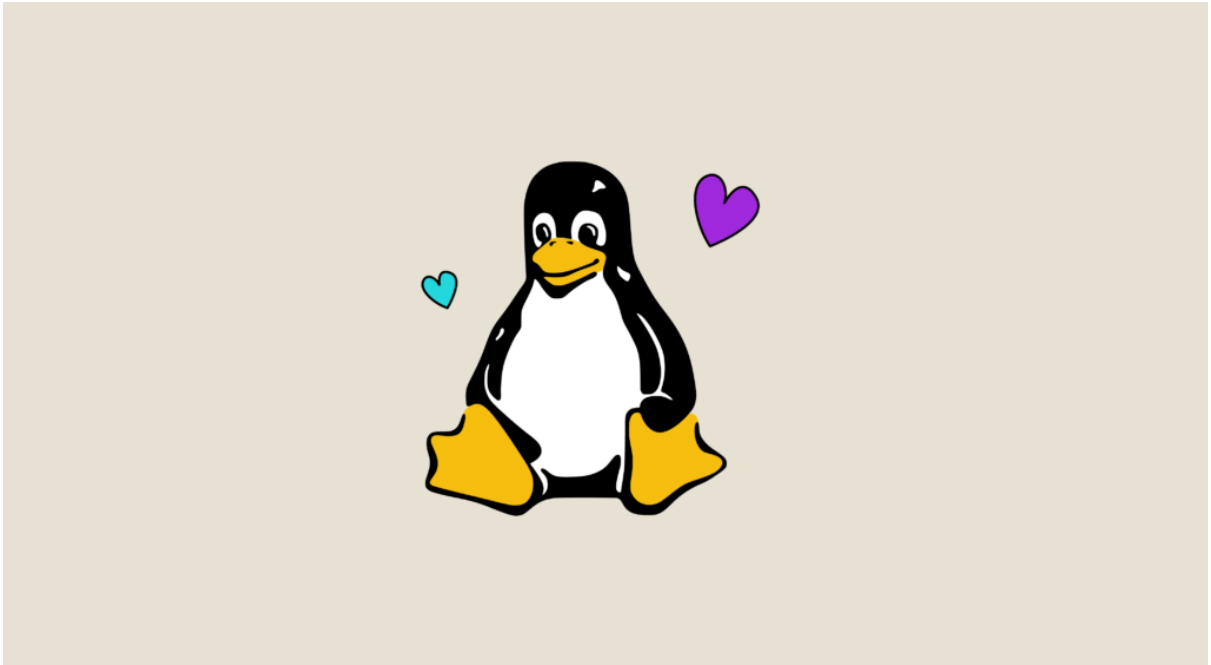
Compte rendu TP1 de Linux embarqué

Valentin Lejars



J’adore Linux<3

1 Prise en main

Séance 1

```
root@DE10-Standard:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       3.0G  1.3G  1.5G  47% /
devtmpfs        375M     0  375M   0% /dev
tmpfs           376M     0  376M   0% /dev/shm
tmpfs           376M  9.8M  366M   3% /run
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           376M     0  376M   0% /sys/fs/cgroup
tmpfs            76M     0   76M   0% /run/user/0
root@DE10-Standard:~#
```

```
Command (m for help): Disk /dev/mmcblk0: 3.7 GiB, 3980394496 bytes, 7774208 sect
ors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x55f3145b

Device         Boot   Start     End Sectors  Size Id Type
/dev/mmcblk0p1         4096 1028095 1024000  500M  b W95 FAT32
/dev/mmcblk0p2      1028096 7774207 6746112  3.2G 83 Linux
/dev/mmcblk0p3         2048    4095    2048    1M a2 unknown

Partition table entries are not in disk order.

Command (m for help): The partition table has been altered.
Calling ioctl() to re-read partition table.
Re-reading the partition table failed.: Device or resource busy

The kernel still uses the old table. The new table will be used at the next rebo
ot or after you run partprobe(8) or kpartx(8).

Please reboot system and execute ./resize2fs_once.
root@DE10-Standard:~#
```

```
root@DE10-Standard:~# ./resize2fs_once
Starting resize2fs_once
resize2fs 1.42.13 (17-May-2015)
Filesystem at /dev/mmcblk0p2 is mounted on /; on-line resizing required
old_desc_blocks = 1, new_desc_blocks = 1
The filesystem on /dev/mmcblk0p2 is now 843264 (4k) blocks long.

The filesystem has been enlarged upon.
root@DE10-Standard:~#
```

```
root@DE10-Standard:~# ipconfig
-bash: ipconfig: command not found
root@DE10-Standard:~# ifconfig
eth0      Link encap:Ethernet  HWaddr ca:e4:6d:9b:d8:45
          inet addr:192.168.88.70  Bcast:192.168.88.255  Mask:255.255.255.0
          inet6 addr: fe80::a21f:1eaf:df37:b869/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:71 errors:0 dropped:0 overruns:0 frame:0
          TX packets:26 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7946 (7.9 KB)  TX bytes:2392 (2.3 KB)
          Interrupt:46

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:164 errors:0 dropped:0 overruns:0 frame:0
          TX packets:164 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:12228 (12.2 KB)  TX bytes:12228 (12.2 KB)

root@DE10-Standard:~#
```

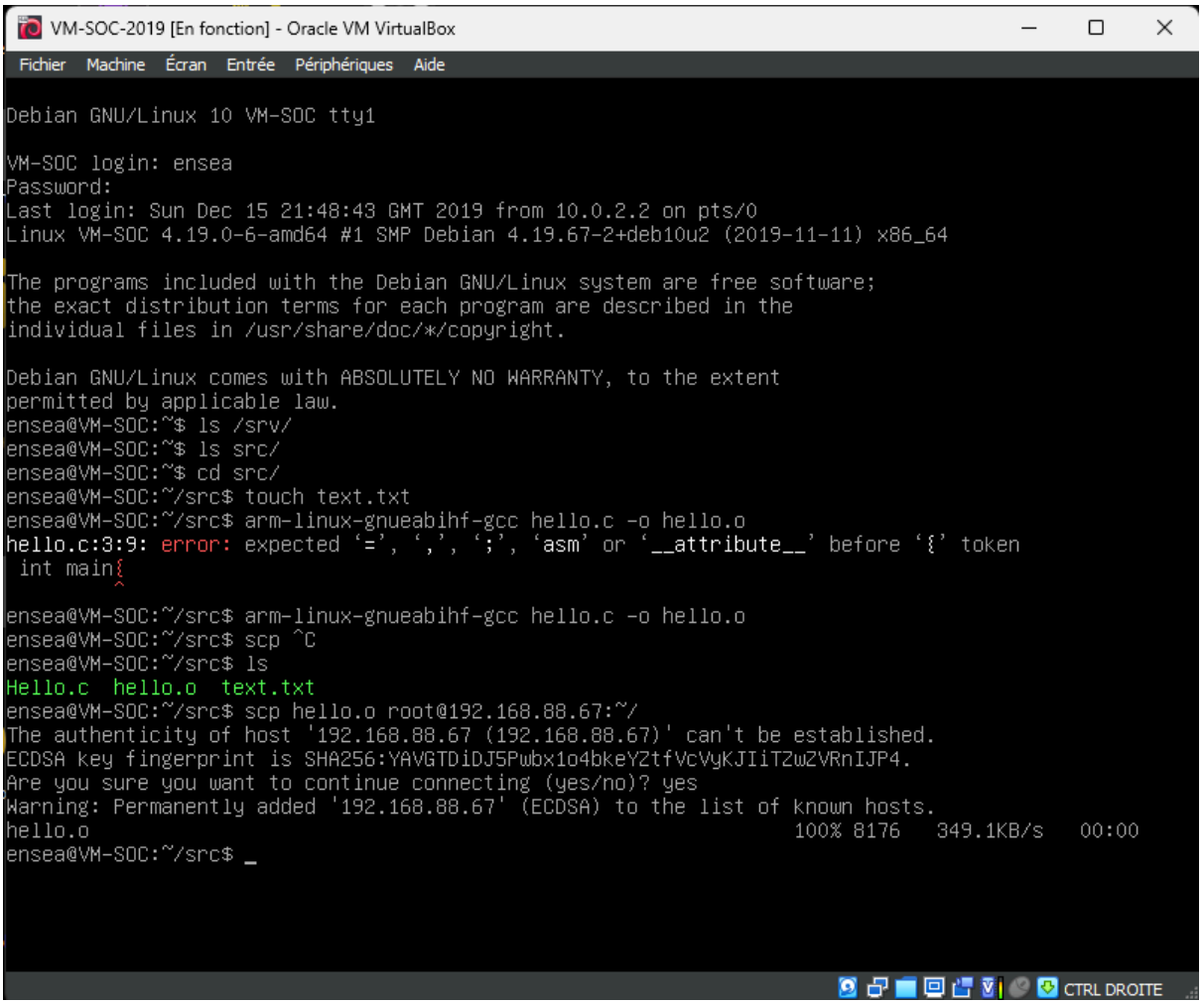
```
root@DE10-Standard:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP gr
oup default qlen 1000
    link/ether 8e:e3:21:42:a2:7e brd ff:ff:ff:ff:ff:ff
    inet 192.168.88.67/24 brd 192.168.88.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::8ce3:21ff:fe42:a27e/64 scope link
       valid_lft forever preferred_lft forever
3: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1
    link/sit 0.0.0.0 brd 0.0.0.0
root@DE10-Standard:~#
```

```
# Logging
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 120
PermitRootLogin yes
StrictModes yes

RSAAuthentication yes
PubkeyAuthentication yes
#AuthorizedKeysFile     %h/.ssh/authorized_keys

# Don't read the user's ~/.rhosts and ~/.shosts files
IgnoreRhosts yes
# For this to work you will also need host keys in /etc/ssh_known_hosts
RhostsRSAAuthentication no
# similar for protocol version 2
HostbasedAuthentication no
# Uncomment if you don't trust ~/.ssh/known_hosts for RhostsRSAAuthentication
#IgnoreUserKnownHosts yes

# To enable empty passwords, change to yes (NOT RECOMMENDED)
PermitEmptyPasswords yes
root@DE10-Standard:~# :wq
-bash: :wq: command not found
root@DE10-Standard:~# ssh root@192.168.88.67
The authenticity of host '192.168.88.67 (192.168.88.67)' can't be established.
ECDSA key fingerprint is SHA256:YAVGTDiDJ5Pubx1o4bkeYZtfVcVyKJIiTZuZVRnIJP4.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.88.67' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.5.0-00198-g6b20a29 armv7l)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
Last login: Wed May 15 13:13:09 2024
root@DE10-Standard:~#
```

On écrit le code hello.c:

```c
#include <stdio.h>

int main(int argc,char *argv[]){
    printf("Hello World\n\r");
    return 0;
}
```

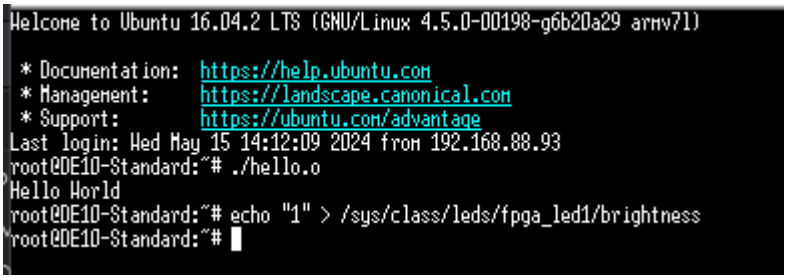
On compile sur la VM puis on envoie le programme sur la carte:



On peut ensuite exécuter le programme sur la carte:

Séance 2

Pour commencer cette nouvelle séance j'ai reconnecté directement la carte à Teraterm via le réseau. Puis j'ai demandé l'exécution du programme hello.c qui avait été réalisé la dernière fois.



La carte répond bien Hello World.
Ensuite j'ai utilisé une commande pour allumer une led sur la carte, celle-ci s'est bien allumée.

Pour le chenillard j'ai écrit un code en c:

```c
#include <stdio.h>
#include <unistd.h>

int main(int argc,char *argv[]){
    int i;
    char str[] = "/sys/class/leds/fpga_led0/brightness";
    FILE * file;

    while(1)
    {
        for(i=0;i<10;i++)
        {
            sprintf(str,"/sys/class/leds/fpga_led%d/brightness",i);
            file = fopen(str,"w");
            fprintf(file,"1");
            fclose(file);
            usleep(100000);
        }
        for(i=0;i<10;i++)
        {
            sprintf(str,"/sys/class/leds/fpga_led%d/brightness",i);
            file = fopen(str,"w");
            fprintf(file,"0");
            fclose(file);
            usleep(100000);
        }
    }
    return 0;
}
```

C'est un chenillard un peu custom qui allume toutes les LEDs une par une puis les éteint une par une. Il fonctionne.

2 Modules kernel

Pour commencer cette deuxième partie je fais à nouveau un chenillard, cette fois ci en utilisant le mmap.

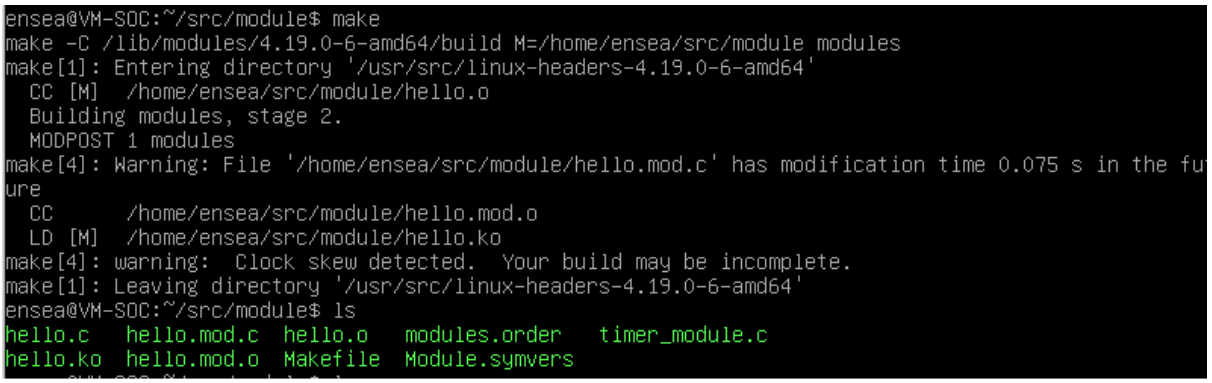
```c
#include <stdio.h>
#include <unistd.h>
#include <stdint.h>
#include <sys/mman.h>
#include <fcntl.h>

int main(int argc,char *argv[]){
    int i;

    while(1)
    {
        for(i=0;i<10;i++)
        {
            uint32_t * p;
            int fd = open("/dev/mem", O_RDWR);
            p = (uint32_t*)mmap(NULL, 4, PROT_WRITE|PROT_READ, MAP_SHARED,
            fd, 0xFF203000);
            *p = (1<<i);
            usleep(10000);
        }
        for(i=0;i<10;i++)
        {
            uint32_t * p;
            int fd = open("/dev/mem", O_RDWR);
            p = (uint32_t*)mmap(NULL, 4, PROT_WRITE|PROT_READ, MAP_SHARED,
            fd, 0xFF203000);
            *p = (0<<i);
            usleep(10000);
        }
    }
    return 0;
}
```

Le code fonctionne mais se coupe au bout d'un moment en renvoyant "segmentation fault". Pour régler ce problème j'ai retiré les déclarations de *p et fd de la boucle while pour les mettre au début du main.

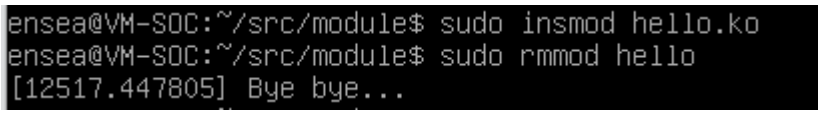
```c
#include <stdio.h>
#include <unistd.h>
#include <stdint.h>
#include <sys/mman.h>
#include <fcntl.h>

int main(int argc,char *argv[]){
    int i;
    uint32_t * p;
    int fd = open("/dev/mem", O_RDWR);

    while(1)
    {
        for(i=0;i<10;i++)
        {
            p = (uint32_t*)mmap(NULL, 4, PROT_WRITE|PROT_READ, MAP_SHARED,
            fd, 0xFF203000);
            *p = (1<<i);
            usleep(10000);
        }
        for(i=0;i<10;i++)
        {
            p = (uint32_t*)mmap(NULL, 4, PROT_WRITE|PROT_READ, MAP_SHARED,
            fd, 0xFF203000);
            *p = (0<<i);
            usleep(10000);
        }
    }
    return 0;
}
```

Problème réglé.

D’abord on make puis on regarde que le hello.ko est bien là.



Puis on fait sudo insmod hello.ko et sudo rmmod hello:



et enfin on regarde dans sudo dmesg: